

Manual de usuario de SiMuL@P (Simulador de Perceptrón Multicapa)

Autores Proyecto: Christian Felipe Álvarez
Javier García-Cuerva Velasco

Proyecto dirigido por: José María Valls Ferrán

Índice

Manual de usuario de SiMuL@P

(Simulador de Perceptrón Multicapa)

En este manual se guiará al usuario en el uso del programa de una manera progresiva, con ejemplos reales y visualizaciones durante la ejecución del programa.

1. Introducción a SiMuL@P

SiMuL@P es un simulador de redes de neuronas, que usa el modelo del Perceptrón Multicapa (MultiLayer Perceptron), para la resolución de determinados problemas no lineales.

Es un programa de uso sencillo y bastante intuitivo concebido como una herramienta para el aprendizaje y el estudio de las redes neuronales artificiales.

2. Perceptrón Multicapa (MLP)

El Perceptrón Multicapa es una generalización del Perceptrón Simple y que fue desarrollado para hacer frente al problema de que este sólo puede resolver problemas lineales. Su diferencia radica en que entre las capas de entrada y de salida se introdujeron una o varias capas ocultas. Este hecho y el uso de la retropropagación hacen que el Perceptrón Multicapa sea una red bastante potente para aproximar funciones.

3. Inicio del programa

Para arrancar el programa necesitamos:

- Tener instalado una versión de Java posterior a la 1.5.
- El archivo simulap.jar que permitirá ejecutar el programa.

Podemos ejecutar simulap.jar de varias formas:

1. Haciendo doble clic sobre el archivo jar o pulsando con el botón derecho sobre el archivo y en el menú contextual seleccionar *abrir con*, eligiendo la versión de Java. Si esto no funcionase utilizaremos la segunda forma.
2. Abrimos un terminal y accedemos al directorio donde se encuentra el archivo jar y escribimos el comando de java:

```
java -jar simulap.jar
```

Después de arrancar el programa se cargará la ventana principal que se muestra en la figura 1 y ya podemos empezar a utilizar SiMuL@P.



Figura 1. Menú principal

4. Empezando a usar el programa

Para poder entrenar a una red de neuronas necesitamos tener la **red de neuronas** y los **patrones** de entrenamiento.

4.1 Empezaremos obteniendo la red neuronal:

La red la podemos **crear**, o bien, **cargar** una red ya creada.

a) Para **crear una red** debemos pinchar sobre el botón de ***Crear nueva red*** y nos aparecerá la ventana de la figura 2.

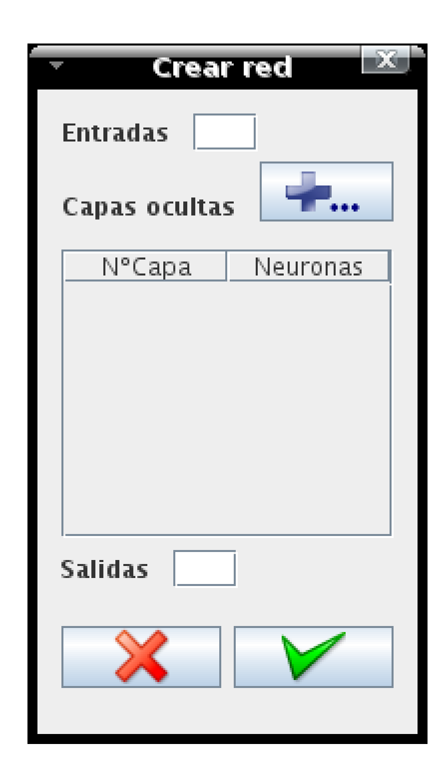


Figura 2. Ventana Crear nueva red

A continuación debemos rellenar los datos de esta ventana para obtener nuestra red:

- En el cuadro de texto de ***Entradas*** debemos escribir el número de neuronas de entrada que queremos que tenga la red, por ejemplo 4 entradas.
- Para ir añadiendo nuevas capas ocultas a la red pincharemos en el botón de ***Capas ocultas*** y nos aparecerá la ventana de la figura 3.

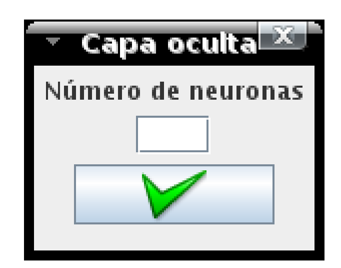


Figura 3. Ventana para añadir una capa oculta

- En esta nueva ventana introduciremos el número de neuronas que queremos que tenga la primera capa oculta, por ejemplo 10. Y para terminar pulsaremos el botón de ***Aceptar***. Esto hará que volvamos a la ventana de la figura 2 con la nueva capa añadida a la tabla de capas ocultas. Para el ejemplo, añadiremos dos nuevas capas ocultas de 10 neuronas también.
- Si cometiésemos un error introduciendo las capas ocultas podemos eliminarlas: seleccionando la capa a eliminar, pulsando sobre ella con el botón derecho del ratón y pulsando sobre ***Borrar***, como muestra la figura 4.

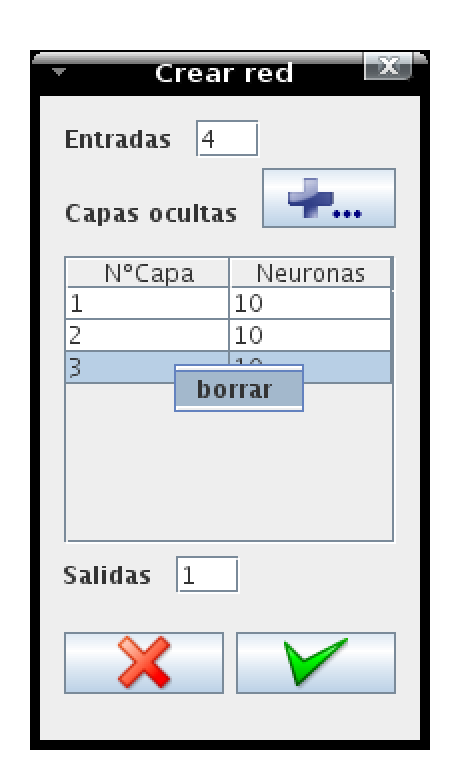


Figura 4. Ventana Crear nueva red eliminado capa

- En el cuadro de texto de ***Salidas*** debemos escribir el número de neuronas de salida que queremos que tenga la red, por ejemplo 1 salida.
- Cuando hayamos rellenado todos los campos y estemos de acuerdo con la red, pulsaremos el botón de ***Aceptar*** . Si en cualquier momento queremos cerrar la ventana de ***Crear nueva red***, perdiendo cualquier posible dato introducido, pulsaremos el botón de ***Cancelar*** .

b) Para **cargar una red** debemos pinchar sobre el botón de ***Abrir red*** y nos aparecerá la ventana de la figura 5, en la cual deberemos elegir el archivo que contiene la red con la que queremos trabajar.



Figura 5. Ventana para elegir el archivo de la red

4.2 El siguiente paso será obtener los <u>patrones de entrenamiento</u> mediante los cuales la red va a aprender:

Para ello, en el menú principal, pincharemos en el botón de ***Seleccionar los ficheros de patrones*** . Nos aparecerá la ventana de la [figura 6](#).

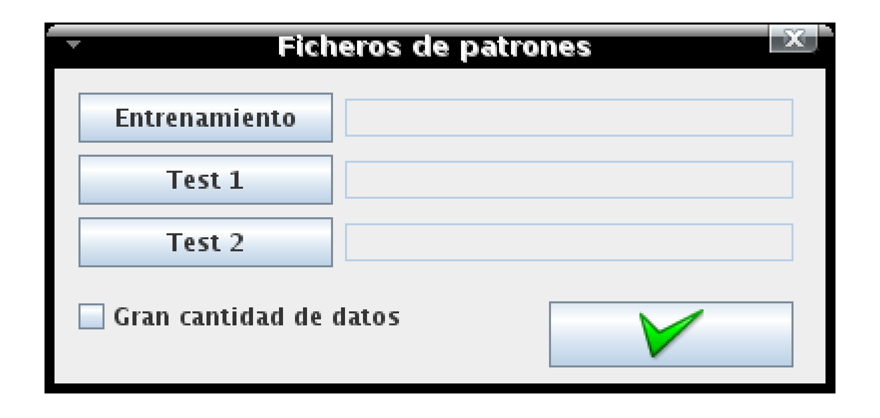


Figura 6. Ventana para seleccionar los ficheros de patrones

Los patrones con los que vayamos a trabajar los debemos separar, en primera instancia, en dos conjuntos, <u>**Entrenamiento**</u> (Train) y <u>**Test**</u> (Test 2):

- El conjunto de entrenamiento se utilizará para entrenar la red.
- El conjunto de test se usará para calcular el error de entrenamiento.

Ya que el Perceptrón Multicapa puede perder la capacidad de generalización, debemos evaluar el error que comete la red durante el proceso de aprendizaje, por esto, existe el conjunto de patrones de validación o Test 1 y sirve para calcular el error de validación y observar que no se produce ningún efecto extraño durante el aprendizaje como el sobreaprendizaje o la parálisis.

Por tanto el conjunto de Entrenamiento será dividido en conjunto de Entrenamiento y conjunto de Test 1 (Conjunto de Validación).

Finalmente tendremos tres ficheros de patrones:

- Conjunto de Entrenamiento (Train). Entrenamiento
- Conjunto de Validación (Test 1). Test 1
- Conjunto de Test (Test 2). Test 2

En la figura 7 podemos ver como se divide el fichero original de patrones.

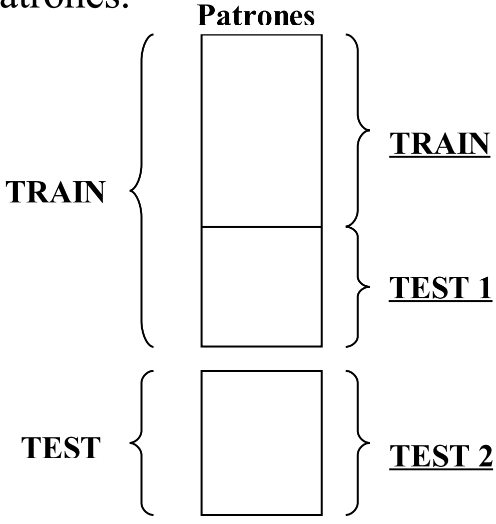


Figura 7. División de patrones

4.3 El formato de los ficheros de patrones debe ser el siguiente:

NÚMERO DE NEURONAS DE ENTRADA
$\mathbf{n_e}$

NÚMERO DE NEURONAS DE SALIDA
$\mathbf{n_s}$

PATRONES (1 patrón en cada línea, la/s entrada/s y la/s salida/s)
x_1 x_2 .... x_{ne} y_1 y_2......... y_{ns}
x_1 x_2 .... x_{ne} y_1 y_2......... y_{ns}
x_1 x_2 .... x_{ne} y_1 y_2......... y_{ns}
....
....

Un fichero de patrones, para el ejemplo de la red creada en el apartado anterior, es el siguiente:

```
#Numero de entradas
4
#Numero de salidas
1
#
# Patrones
#
0.223682776 0.328490116 0.384189034 0.413789723 0.421620155

0.24605079  0.340377396 0.390506422 0.421620155 0.43269275

0.266182002 0.351075947 0.396192071 0.43269275  0.4462798
```

Los caracteres '#' indican comentarios.

Cuando se tengan divididos los patrones en los tres ficheros correspondientes, en la ventana correspondiente a la figura 6, pincharemos en cada botón: ***Entrenamiento, Test 1* y *Test 2***, y seleccionaremos en la ventana de abrir archivo (misma ventana que figura 5) el fichero correspondiente.

En la parte inferior de la ventana de la figura 6, existe la posibilidad de seleccionar si estamos usando una ***Gran cantidad de datos*** Gran cantidad de datos, es decir, si los patrones son muy numerosos, para no cargar todos en memoria principal, leeremos varias veces del fichero del disco, si esta opción esta marcada. El programa necesitará más tiempo para ejecutar. Si esta opción no esta marcada y el número de patrones no es excesivo, el programa cargará los datos en memoria principal, evitando demasiados accesos a disco y que por tanto el tiempo de ejecución se vea incrementado. En la mayoría de los casos esta opción deberá estar desactivada.

Cuando hayamos seleccionado todos los ficheros de patrones, pulsaremos el botón de ***Aceptar*** .

5. Entrenando la red

Una vez tenemos la red de neuronas definida, ya podemos definir los parámetros que necesitamos para que la red aprenda de los patrones.

Pulsaremos el botón de ***Entrenamiento*** y en la parte inferior de la ventana (ver figura 8) nos aparecerán los campos que necesitamos rellenar para definir como va a ser el entrenamiento.



Figura 8. Ventana de Entrenamiento

A la izquierda podemos definir el ***Rango*** Rango de inicialización, sobre el que se inicializarán los Pesos y Umbrales. Pulsando el botón de ***Inicializar pesos y umbrales*** , se inicializarán estos, de manera aleatoria entre el rango de inicialización fijado.

En la parte derecha podremos definir el número de ***Ciclos*** Ciclos, que queremos que tenga el entrenamiento de la red completa, la cual usará el fichero de patrones. También podemos indicar cada cuantos ciclos de entrenamiento queremos que se usen

los patrones de validación o de Test 1 (***Ciclos-Test*** Ciclos-Test), los cuales ayudarán a calcular el error de validación. El último parámetro necesario es la ***Tasa o razón de aprendizaje*** Tasa de aprendizaje, el cual influye en la velocidad de convergencia hacia la función que se desea y que debe ser un valor bajo.

Una vez tengamos todos los parámetros definidos podemos iniciar el entrenamiento con el botón de ***Entrenar*** .

A continuación se cargará la ventana con la gráfica que muestra cómo entrena la red (figura 9), en la que se ve el error de entrenamiento y el error de test 1.

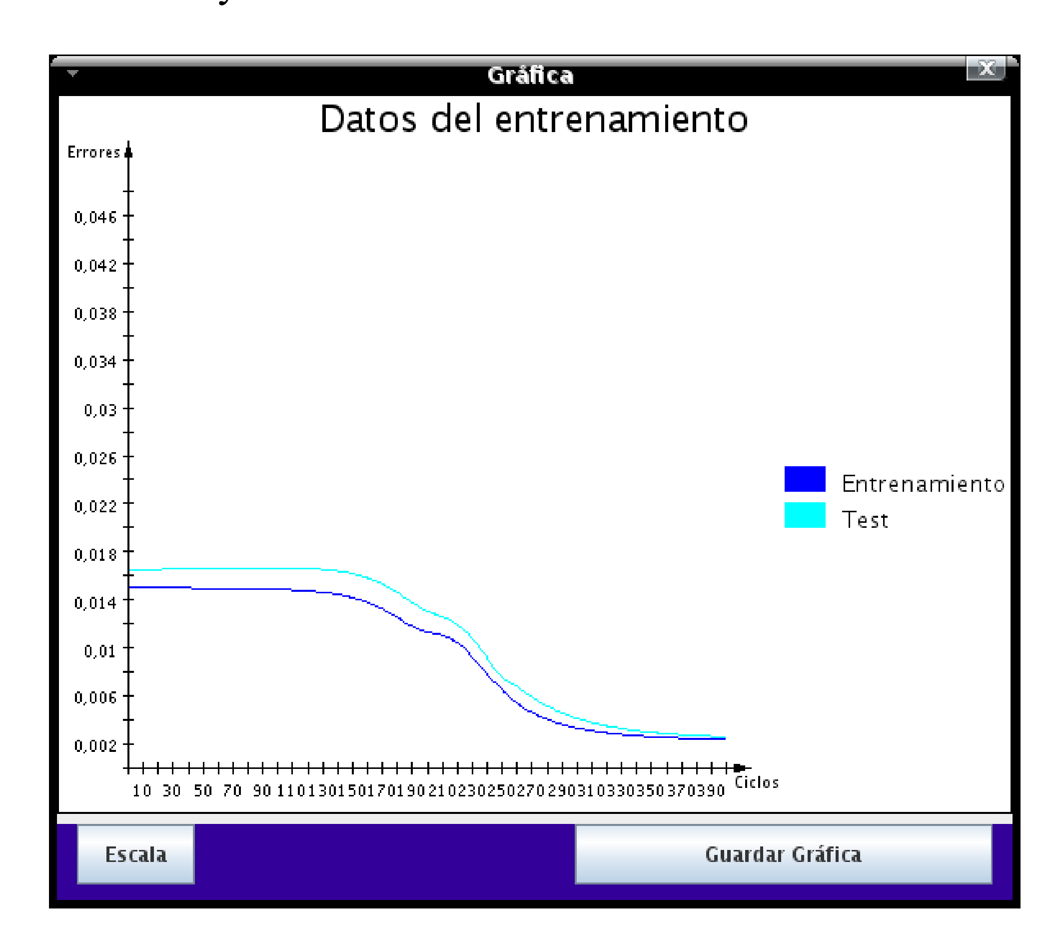


Figura 9. Ventana con la gráfica de entrenamiento

Durante el entrenamiento podemos variar la escala en la que se representan los datos, para ello pulsaremos el botón ***Escala*** Escala y a continuación nos aparecerá la ventana para poder variar la misma (figura 10).



Figura 10. Diálogo para cambiar la escala de la gráfica

Para cambiar la escala, iremos incrementando o decrementando los valores que delimitan el rango del error cometido (representado en el eje Y). Los valores cambiarán de 0.05 en 0.05 utilizando los botones (+, -).

Cuando ha terminado de entrenar los ciclos que se han indicado se pueden cambiar los parámetros que definen el entrenamiento, aunque para que el entrenamiento sea coherente, sólo se deberá cambiar el número de ciclos (entrenamiento y/o test 1). Pulsando sucesivas veces el botón de ***Entrenar*** se continúa con el entrenamiento de la red los ciclos que se haya indicado.

Cuando estemos conformes con la gráfica o queramos conservarla podemos guardarla mediate el botón ***Guardar gráfica*** Guardar Gráfica . Habrá que indicar un nombre al archivo, sin extensión, ya que esta viene determinada por el formato de la imagen (PNG: *Portable Network Graphics*).

Si en cualquier momento deseamos finalizar el entrenamiento le daremos al botón de ***Finalizar entrenamiento***. Al finalizar el entrenamiento se conservan en memoria los pesos y umbrales que se han determinado durante el mismo. Si el entrenamiento ha resultado satisfactorio será recomendable guardar la red entrenada (Ver punto 6).

5.1 Los <u>datos de entrenamiento</u> se guardan en un fichero con el siguiente formato:

# DATOS DEL ENTRENAMIENTO		
NÚMERO_CICLO	ERROR_MEDIO	ERROR_ACUMULADO
$\mathbf{N_1}$	$\mathbf{E_{M1}}$	$\mathbf{E_{A1}}$
$\mathbf{N_2}$	$\mathbf{E_{M2}}$	$\mathbf{E_{A2}}$
$\mathbf{N_3}$	$\mathbf{E_{M3}}$	$\mathbf{E_{A3}}$

Donde $\mathbf{N_i}$ indica el número de ciclo, $\mathbf{E_{Mi}}$ representa el error medio para el ciclo *i*, y $\mathbf{E_{Ai}}$ es el error acumulado para el ciclo *i*.

El fichero se guardará en el mismo directorio que el fichero de patrones de entrenamiento utilizado con el mismo nombre que éste, salvo porque le añade el sufijo “.res”.

Los resultados de test-1 se guardan en otro fichero al que se le añade el sufijo “.res.test1” y que tiene el siguiente formato:

# DATOS DE TEST1		
NÚMERO_CICLO	ERROR_MEDIO	ERROR_ACUMULADO
$\mathbf{N_1}$	$\mathbf{E_{M1}}$	$\mathbf{E_{A1}}$
$\mathbf{N_2}$	$\mathbf{E_{M2}}$	$\mathbf{E_{A2}}$
$\mathbf{N_3}$	$\mathbf{E_{M3}}$	$\mathbf{E_{A3}}$

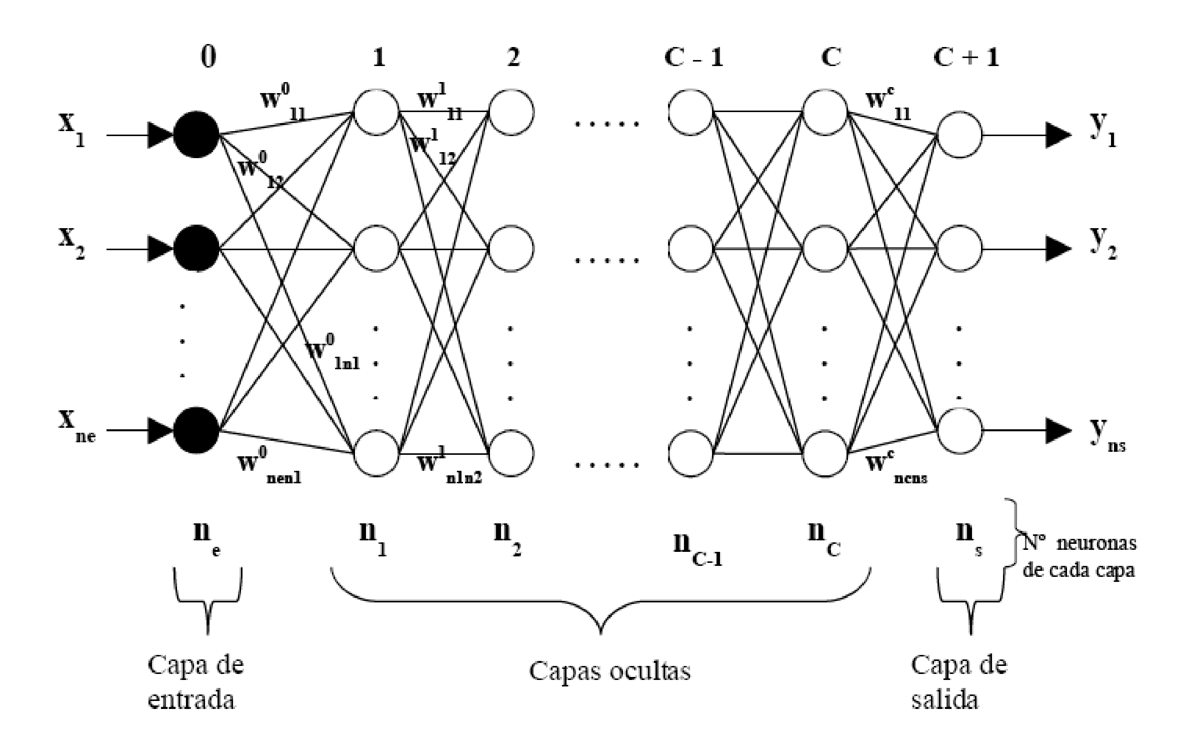
El número de ciclo indica en que ciclo se ha hecho el test-1.

6. Guardando la red

En cualquier momento podemos desear guardar la red, es decir, con la entrenada o no, para poder cargarla en otro momento y seguir trabajando con ella.

Para guardar la red pulsaremos sobre el botón de ***Guardar red***, y nos aparecerá una nueva ventana donde podremos elegir el directorio donde guardarla.

La arquitectura de la red del Perceptrón Multicapa es la siguiente:



Pesos: $\mathbf{w}^{c}_{ij}$ (Peso de la conexión de la neurona i de la capa c hasta la neurona j de la capa c+1)
Umbrales: $\mathbf{u}^{c}_{i}$ (Umbral de la neurona i de la capa c (excepto la 0))

Esta red se guarda en un fichero de texto con el siguiente formato:

NOMBRE DEL FICHERO
FECHA

NÚMERO DE NEURONAS DE ENTRADA
$\mathbf{n_e}$

NÚMERO DE CAPAS OCULTAS
$\mathbf{c}$

NEURONAS POR CADA CAPA:
$\mathbf{n_1 \quad n_2 \ldots\ldots\ldots\ldots n_c}$

NÚMERO DE NEURONAS DE SALIDA
$\mathbf{n_s}$

PESOS
$\mathbf{w^0_{11} \quad w^0_{12} \ldots\ldots w^0_{1n1} \quad w^0_{21} \ldots\ldots w^0_{nen1}}$
$\mathbf{w^1_{11} \quad w^1_{12} \ldots\ldots w^1_{1n2} \quad w^1_{21} \ldots\ldots w^1_{n1n2}}$
....
$\mathbf{w^c_{11} \quad w^c_{12} \ldots\ldots w^c_{1ns} \quad w^c_{21} \ldots\ldots w^c_{ncns}}$

UMBRALES
$\mathbf{u^1_1 \quad u^1_2 \ldots\ldots u^1_{n1}}$
$\mathbf{u^2_1 \quad u^2_2 \ldots\ldots u^2_{n2}}$
....
$\mathbf{u^c_1 \quad u^c_2 \ldots\ldots\ldots u^c_{nc}}$
$\mathbf{u^{c+1}_1 \quad u^{c+1}_2 \ldots\ldots u^{c+1}_{ns}}$

Para una red de 2 entradas, con dos capas ocultas de 3 neuronas cada una y una salida; el fichero, con los pesos y umbrales inicializados a 0, sería el siguiente:

```
# Nombre del fichero: ejemplo.red
# Creado el 21/01/2007

# Fichero en el que se ha guardado la red cuya estructura es:

# Numero de entradas
2
# Numero de capas ocultas
2
# Neuronas por cada capa
3     3
# Numero de salidas
1
# Pesos
0.0   0.0   0.0   0.0   0.0   0.0
0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
0.0   0.0   0.0
# Umbrales
0.0   0.0   0.0
0.0   0.0   0.0
0.0
```

Los caracteres '#' indican comentarios.

7. Calculando test 2

Una vez tenemos la red entrenada podemos proceder a obtener los datos del test 2. Estos consisten en la *salida obtenida*, la *salida deseada*, el *error absoluto* y el *error cuadrático* para cada patrón de test, además de los *errores acumulados y medios*.

Pulsando en el botón ***Test-2*** se nos pide el nombre del fichero en el que se guardarán, éste se almacenará en el mismo directorio que el fichero que contiene los patrones de test 2.

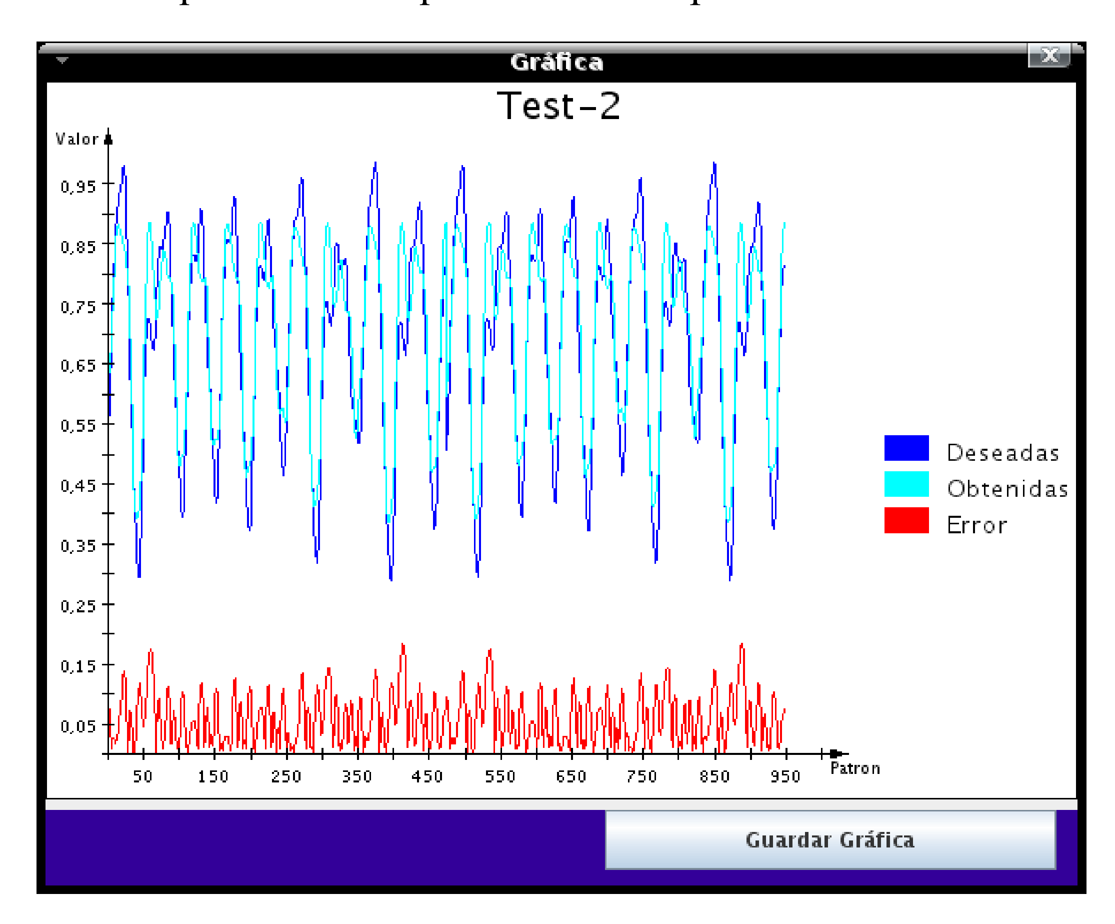


Figura 11. Gráfica de test 2

Habiendo aceptado aparece la ventana en la que se representan la salida obtenida, la salida deseada y el error absoluto en una gráfica (figura 11). Si se desea guardar esta gráfica se deberá pulsar el botón ***Guardar gráfica*** tal como en las gráficas de entrenamiento.

7.1 Los datos de test 2 se guardan en un fichero con el siguiente formato:

```
# DATOS DE LA VALIDACIÓN

N_PATRÓN   SALIDA_OBTENIDA   SALIDA_DESEADA   ERROR_ABSOLUTO   ERROR_CUADRÁTICO
P1         Y1                D1               EA1              EC1
P2         Y2                D2               EA2              EC2

                                 (...)
--------------------------------------------------------------------------------
ERROR_ABSOLUTO_ACUMULADO: EAA
ERROR_ABSOLUTO_MEDIO: EAM
ERROR_CUADRÁTICO_ACUMULADO: ECC
ERROR_CUADRÁTICO_MEDIO: ECM
```